Backdoor Removal by Task Negation in Weight Space

Anonymous authors
Paper under double-blind review

Abstract

Foundation models have revolutionized computer vision by enabling broad generalization across diverse tasks. Yet, they remain highly susceptible to targeted backdoor attacks. Mitigating such vulnerabilities remains an open challenge, especially given that the large-scale nature of these models prohibits retraining to ensure safety. Existing backdoor removal approaches rely on costly fine-tuning to override the harmful behavior, and can often degrade performance on other unrelated tasks. This raises the question of whether backdoors can be removed without compromising the general capabilities of the models. In this work, we address this question and study how backdoors are encoded in the model weight space, and show that they are approximately *disentangled* from other benign tasks. Specifically, this separation enables the isolation and erasure of the backdoor's influence on the model with minimal impact on clean performance. Building on this insight, we introduce a simple post-hoc removal method that leverages such disentanglement. Through extensive experiments with CLIP-based models and common adversarial triggers, we show that, given a small triggered set, our method reduces attack success rate by over 99% while retaining, on average, 96% of the model's clean accuracy, using less than 2% of the data required by clean-data fine-tuning defenses. We further find that trigger vectors estimated on one dataset transfer to backdoored models trained on other datasets and even other label spaces, indicating that standard attacks induce a consistent direction in weight space. Additionally, we demonstrate that even when the attack and its presence are unknown, our method can be paired with reverse-engineered triggers to mitigate the effects of the backdoor. Overall, our method offers favorable removal and clean-accuracy tradeoffs compared to state-of-the-art clean data defenses, at a fraction of their data cost.

1 Introduction

Foundation models have become a cornerstone of modern deep learning, offering broad generalization across a wide range of tasks through large-scale pre-training (Radford et al., 2021; Jia et al., 2021). Among them, vision-language models like CLIP (Radford et al., 2021) play a fundamental role. They not only demonstrate remarkable robustness to distribution shifts and *zero-shot* performance on out-of-distribution benchmarks (Wortsman et al., 2022b), but their vision encoders also serve as a key component in many multimodal large language models, such as LLaVA (Liu et al., 2023).

However, the very success and widespread integration of these models make them a prime target for security threats, most notably *backdoor attacks* (Carlini & Terzis, 2022; Bansal et al., 2023; Huang et al., 2025) – a class of threats that compromise model integrity even after training is complete. In a backdoor attack (Gu et al., 2017), an adversary poisons a small portion of the training data by embedding a fixed trigger pattern into inputs and mislabeling them to a target class. The resulting model appears to perform well on clean inputs but systematically misclassifies any input containing the trigger – effectively granting the adversary precise control over model predictions. Such vulnerabilities pose a serious risk in safety-critical applications, including autonomous driving and medical diagnostics (Du et al., 2024; Hanif et al., 2024).

Current defenses for CLIP largely fall into two categories: *(i)* retraining the model from scratch using modified loss functions designed to resist backdoors, or *(ii)* fine-tuning on clean data to override the malicious behavior (Bansal et al., 2023; Yang et al., 2024; Goel et al., 2022a; Zhang et al., 2025). However,

full retraining is prohibitively expensive at scale, while fine-tuning – though cheaper – frequently induces *catastrophic forgetting* (French, 1999), whereby the pre-trained knowledge is erased. Furthermore, recent studies show that fine-tuning strategies struggle against more sophisticated attacks (Liang et al., 2024).

An alternative line of work, spanning approximate machine unlearning (Cao & Yang, 2015) and post-hoc model editing, seeks to selectively remove specific learned behaviors post-training, avoiding full retraining. Currently, the application of unlearning methods to targeted backdoor removal remains limited. Prominent optimization-based algorithms such as *gradient ascent* and its variants have been shown to fall short when applied to backdoor removal in small-scale settings (Pawelczyk et al., 2025). Yet, their effectiveness in large-scale foundation models remains an open question.

To overcome these limitations, we turn to recent advances in model editing in weight space (Frankle et al., 2020; Izmailov et al., 2018; Wortsman et al., 2021; 2022a; Rame et al., 2022; Ainsworth et al., 2023; Ilharco et al., 2022; 2023). In particular, Ilharco et al. (2023) introduced the concept of a *task vector*, which is the element-wise difference between the weights of a fine-tuned model and its pre-trained initialization. Task vectors provide a means to encode learned tasks as directions in weight space. They can be added to a model to inject functionality, subtracted to unlearn specific tasks, or combined to compose multi-task models. These manipulations are enabled by the *disentanglement* of tasks in the weight space of pre-trained models, as recently formalized by Ortiz-Jimenez et al. (2024).

In this paper, we build on this framework to approach backdoor removal in CLIP-based models from a more fundamental perspective and study whether the backdoor mechanism is *geometrically separable* from the rest of what the model has learned. Indeed, if the parameter update encoding the backdoor occupies a direction whose functional effect is approximately disjoint from the directions supporting clean capabilities, then removal reduces to estimating that direction and subtracting it.

By treating triggered behavior as a task-vector direction in weight space, we empirically show that the weights of backdoored models can be linearly decomposed into approximately clean and triggered components, effectively disentangling the malicious behavior from the model's benign capabilities. This disentanglement allows us to isolate and remove the backdoor's influence by exploiting task arithmetic. In practice, this is achieved by fine-tuning the model on a small set of triggered examples to compute a "trigger vector". This vector can then be subtracted – via task negation – to surgically remove the backdoor while preserving clean model performance, as illustrated in Figure 1. Interestingly, in controlled image-classification experiments, we find evidence that a trigger-vector direction estimated from one backdoored model can be re-used to sanitize another backdoored model trained on a different dataset, enabling transfer under a shared trigger, which reinforces the notion that the vector primarily encodes for the malicious behavior.

Our main contributions are:

- **Weight disentanglement of backdoors** We provide empirical evidence that for common backdoor attacks on CLIP the parameter updates leading to a backdoored model approximately decompose into a clean and triggered direction that act additively in function space. This decomposition enables target removal via linear operations in weight space without catastrophic forgetting of non-adversarial knowledge.

- **Method** We introduce `TBAR` (**T**rigger removal by **B**ackdoor **AR**ithmetic), a lightweight model editing approach for backdoor removal via weight-space task negation. In scenarios where the trigger is known, `TBAR` removes 99% of the backdoor while retaining 96% of obtained clean accuracy on average across *(i)* image classification backdoor benchmarks and *(ii)* large-scale image-captioning tasks. Notably, in the latter case, it outperforms state-of-the-art clean-data fine-tuning defenses while using less than 2% of the data requirements.

- **Transferability of trigger directions** A trigger vector estimated from a backdoored ImageNet model removes the same trigger from CIFAR- and SUN-trained models – even when the target label differs – indicating that standard attacks induce a consistent, reusable direction in weight space rather than dataset-specific associations.

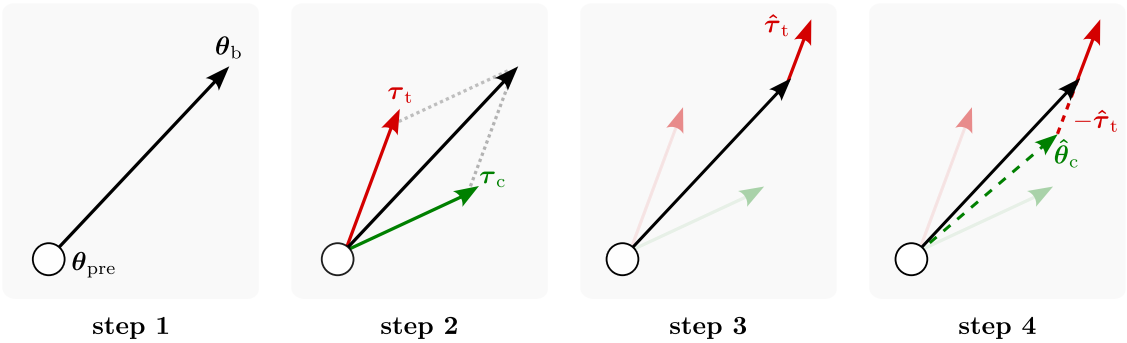

Figure 1: Backdoored models embed malicious behavior along with clean task performance. We propose a targeted approach: (1) Given a backdoored model $\boldsymbol{\theta}_{\mathrm{b}}$, fine-tuned from a base model $\boldsymbol{\theta}_{\mathrm{pre}}$. (2) The backdoor encodes two distinct directions: clean $\boldsymbol{\tau}_{\mathrm{c}}$ and triggered $\boldsymbol{\tau}_{\mathrm{t}}$. (3) Fine-tuning the model on triggered data isolates the parameter shift associated with triggered information, yielding an estimated trigger vector $\hat{\boldsymbol{\tau}}_t$. (4) Negating this vector from the original parameters effectively removes the trigger to obtain a sanitized $\hat{\boldsymbol{\theta}}_{\mathrm{c}}$ while preserving clean task performance.

- **Connection to gradient ascent** We find that plain gradient ascent on triggered examples is also effective at this scale, in contrast with small-scale reports (Pawelczyk et al., 2025), and argue that this success is a consequence of weight disentanglement. In particular, `TBAR` can be viewed as a stable, controlled instantiation of the same underlying edit, replacing gradient ascent's fragile stopping criteria with a single bounded scaling coefficient.
- **Trigger-unknown removal** In settings where the trigger is unknown, we extend `TBAR` to operate in an attack-agnostic scenario by pairing it with reverse-engineered proxy triggers. Our method sanitizes infected models for all attacks for which a usable trigger can be recovered, outperforming state-of-the-art defenses while preserving over 90% clean accuracy.

2 Problem Setup

This work focuses on security vulnerabilities associated with backdoor attacks. Specifically, we consider the following threat model and defender assumptions. This setup is an extension of several previous settings in the literature (Carlini & Terzis, 2022; Bansal et al., 2023; Feng et al., 2023; Pawelczyk et al., 2025).

Threat model The adversary has full white-box access to a pre-trained model and fine-tuning data to be used to backdoor the model. The attack is carried out by injecting a small poisoned subset into a larger training dataset. The resulting backdoored model is released publicly and intended for downstream use by unaware users. Unless otherwise specified, we consider the attack successful if the triggered examples are predicted as a targeted label.

Defender assumptions The defender's goal is to remove the backdoor (i.e., reduce attack success rate to zero) while preserving the model's performance on clean data. The defender has full access to model weights. We consider two distinct practical scenarios:

- **Trigger-known:** The defender has access to a small set containing the trigger.
- **Trigger-unknown:** The defender does not know the true trigger but has access to a small set of clean data.

The former, trigger-known scenario represents a downstream user who discovers the trigger post-release (e.g., via incident response, red-teaming, or external disclosure) and must patch the checkpoint without

access to the training pipeline. This is also the standard assumption in the post-hoc unlearning literature on poisoning (Pawelczyk et al., 2025). The latter, trigger-unknown, scenario instead corresponds to a defender who suspects the model is compromised but has no knowledge of the trigger itself. Notice that in this setting our results are dependent on the success of the trigger inversion. In particular, the focus of this work is on backdoor removal given a backdoor signal, not on trigger discovery.

3 Background

This section introduces the tools for understanding model editing using weight interpolation. In particular, we recall the operations of task arithmetic and the property of weight disentanglement enabling them.

Despite the non-linear nature of neural networks, previous work has shown that it's possible to interpolate between the weights of two models under certain conditions (Izmailov et al., 2018; Frankle et al., 2020; Wortsman et al., 2021; 2022a; Ainsworth et al., 2023; Ilharco et al., 2022; Wortsman et al., 2022b). This property, alongside a growing interest in multi-task learning, gave rise to the development of weight-space model editing techniques (Wortsman et al., 2022a; Ilharco et al., 2023; Matena & Raffel, 2022; Yadav et al., 2023; Dimitriadis et al., 2023).

Notation Let a neural network be a parameterized function $f : \mathcal{X} \times \Theta \to \mathcal{Y}$ with inputs $x \in \mathcal{X}$ and weights $\boldsymbol{\theta} \in \Theta$. We identify a task $k \in [K]$ as a triplet $(\mathcal{D}_k, \mu_k, f_k^\star)$ with domain $\mathcal{D}_k \subseteq \mathcal{X}$, input distribution μ_k and $f_k^\star : \mathcal{D}_k \to \mathcal{Y}$, with $\mathcal{D}_k = \text{supp}(\mu_k)$.

Model editing with task arithmetic (Ilharco et al., 2023) Fine-tuning a pre-trained model $\boldsymbol{\theta}_{\text{pre}}$ on task k yields new weights $\boldsymbol{\theta}_k^\star$. The change in weights, $\boldsymbol{\tau}_k = \boldsymbol{\theta}_k^\star - \boldsymbol{\theta}_{\text{pre}}$, defines the *task vector*. Task arithmetic modifies the model by applying scaled task vectors: $\boldsymbol{\theta}_{\text{new}} = \boldsymbol{\theta}_{\text{pre}} + \alpha\, \boldsymbol{\tau}_k$ for a single task, or $\boldsymbol{\theta}_{\text{new}} = \boldsymbol{\theta}_{\text{pre}} + \sum_{k=1}^{K} \alpha_k\, \boldsymbol{\tau}_k$ for multiple tasks. The scalar coefficient α controls the strength of the edit as well as its direction, where positive values increase performance on a particular task and negative values degrade it. Similar to other weight interpolation techniques, the optimal value for the scaling coefficient α is typically selected using a small validation set (Ilharco et al., 2022; 2023; Yadav et al., 2023; Ortiz-Jimenez et al., 2024; Hazimeh et al., 2024).

Weight disentanglement To explain task arithmetic, Ortiz-Jimenez et al. (2024) introduced *weight disentanglement* as the property where the functional changes induced by a set of task vectors are localized to their respective task domains. Specifically, when multiple task vectors are linearly combined in weight space, the resulting model behaves as if it selectively applies each individual task's function only for inputs within that task's domain, reverting to the pre-trained model's behavior otherwise. The ability to perform task arithmetic with a set of task vectors $\mathcal{T}$ is a direct consequence of this weight disentanglement, where each task vector $\boldsymbol{\tau}_k$ encodes a distinct functional component specific to its domain $\mathcal{D}_k$. Formally, for a set of task vectors $\{\boldsymbol{\tau}_k\}_{k\in[K]}$, the edited model satisfies weight disentanglement if (cf. Ortiz-Jimenez et al. (2024) for the formal definition):

$$f\left(x; \boldsymbol{\theta}_{\text{pre}} + \sum_{k=1}^{K} \alpha_k \boldsymbol{\tau}_k\right) = \sum_{k=1}^{K} f(x; \boldsymbol{\theta}_{\text{pre}} + \alpha_k \boldsymbol{\tau}_k)\ \mathbb{1}(x \in \mathcal{D}_k) + f(x; \boldsymbol{\theta}_{\text{pre}})\ \mathbb{1}\left(x \notin \bigcup_{k=1}^{K} \mathcal{D}_k\right). \tag{1}$$

To measure the presence of weight disentanglement, Ortiz-Jimenez et al. (2024) introduced the weight disentanglement error, which measures the prediction disagreement between models obtained by applying the individual task vectors and the combination thereof, evaluated on the respective task supports. For two tasks, this reads:

$$\xi(\alpha_1, \alpha_2) = \sum_{i\in\{1,2\}} \mathbb{E}_{x\sim\mu_i}\left[\text{dist}\left(f(x; \boldsymbol{\theta}_{\text{pre}} + \alpha_i \boldsymbol{\tau}_i),\ f(x; \boldsymbol{\theta}_{\text{pre}} + \alpha_1 \boldsymbol{\tau}_1 + \alpha_2 \boldsymbol{\tau}_2)\right)\right], \tag{2}$$

where dist can be any distance metric between model outputs. For instance, for classification tasks $\text{dist}(y_1, y_2) = \mathbb{1}(y_1 \neq y_2)$.

4 TBAR: Trigger Removal by Backdoor Arithmetic

In this section, we leverage the principles of task arithmetic and weight disentanglement to design a method capable of surgically removing backdoor triggers from compromised models. We treat the benign task and the malicious backdoor behavior as two separate – and, ideally, separable – tasks operating on distinct domains, i.e., clean or triggered data. We then present empirical evidence validating the success of this approach.

4.1 Weight disentanglement of clean and triggered tasks

Consider a model with pre-training weights $\boldsymbol{\theta}_{\text{pre}}$ that has been backdoored, resulting in weights $\boldsymbol{\theta}_b$. Let $\boldsymbol{\tau}_c$ and $\boldsymbol{\tau}_t$ be the task vectors for the clean and triggered tasks, respectively, with domains $\mathcal{D}_c$ (clean images) and $\mathcal{D}_t$ (triggered images). Following the definition in Equation 1, the backdoored model satisfies weight disentanglement with respect to these vectors if, $\forall x \in \mathcal{D}_c \cup \mathcal{D}_t$,

$$f(x; \boldsymbol{\theta}_{\text{pre}} + \alpha_c \boldsymbol{\tau}_c + \alpha_t \boldsymbol{\tau}_t) = f(x; \boldsymbol{\theta}_{\text{pre}} + \alpha_c \boldsymbol{\tau}_c)\ \mathbb{1}(x \in \mathcal{D}_c) + f(x; \boldsymbol{\theta}_{\text{pre}} + \alpha_t \boldsymbol{\tau}_t)\ \mathbb{1}(x \in \mathcal{D}_t). \tag{3}$$

In this work, we formulate the following hypothesis:

Hypothesis. *The weights of backdoored vision foundation models satisfy weight disentanglement with respect to the clean and triggered tasks for common backdoor attacks, i.e., their output function f satisfies Equation 3.*

The key implication of this hypothesis is the existence of a specific direction in weight space, $\boldsymbol{\tau}_t$, that predominantly governs the backdoor's malicious behavior. If this holds, removing the backdoor without causing catastrophic forgetting is possible: one needs to estimate $\boldsymbol{\tau}_t$ and subtract it from the model's weights. As we will demonstrate, in practical settings, this approximately holds and allows us to effectively remove the backdoor without compromising the model's clean knowledge.

Specifically, we define a small *trigger set*, disjoint from the original poisoned data, composed entirely of triggered image-target pairs. We fine-tune the backdoored model $\boldsymbol{\theta}_b$ on this set, yielding updated weights $\boldsymbol{\theta}_{b+t}$. The parameter difference from this step gives us an estimate of the trigger direction:

$$\hat{\boldsymbol{\tau}}_t = \boldsymbol{\theta}_{b+t} - \boldsymbol{\theta}_b \tag{4}$$

The presence of this weight disentanglement would then allow us to surgically remove the backdoor's influence from the original backdoored model via task negation, yielding a cleaned model $\hat{\boldsymbol{\theta}}_c$:

$$\hat{\boldsymbol{\theta}}_c = \boldsymbol{\theta}_b - \alpha \hat{\boldsymbol{\tau}}_t \tag{5}$$

where α is a scalar coefficient controlling the strength of the removal. We call this method **T**rigger removal by **B**ackdoor **AR**ithmetic, or TBAR.

A minimal model of backdoor learning To build intuition for TBAR and the underlying decomposition, we consider a minimal model where it naturally emerges. Let $f(x; W) = Wx$ be a linear classifier whose clean inputs lie in a subspace U and let the trigger be a fixed unit vector $p \perp U$, so that a triggered input is $\tilde{x} = z + p$ with $z \in U$. This stylizes patch and blend triggers, which occupy input directions that natural images do not. Because under gradient descent, every gradient update of W is an outer product with the input, training only moves weights along data-spanned directions. Pre-training on clean data therefore preserves trigger-blindness: if at initialization $W_{\text{init}}\, p = 0$, then $W_{\text{pre}}\, p = 0$. Poisoned fine-tuning produces an update that splits exactly into orthogonal components,

$$W_b \;=\; W_{\text{pre}} \;+\; \underbrace{M}_{\substack{\text{clean} \\ Mp=0}} \;+\; \underbrace{v\,p^{\top}}_{\text{trigger}}, \tag{6}$$

where M collects updates along clean-image directions and v accumulates the gradient contributions of poisoned samples toward the target class. On clean inputs, the trigger component vanishes, $W_b z = (W_{\text{pre}} + M)z$, while on triggered inputs it acts as an input-independent offset, $W_b(z + p) = (W_{\text{pre}} + M)z + v$. Any

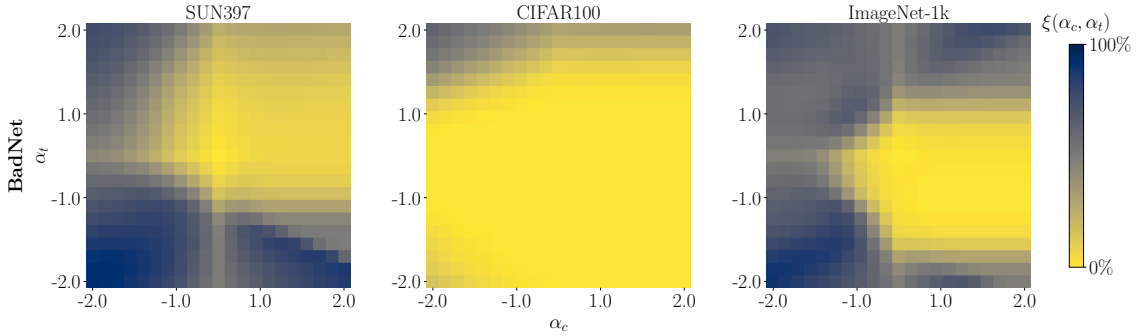

Figure 2: Weight disentanglement between clean and triggered tasks. We estimate the triggered direction $\hat{\boldsymbol{\tau}}_t$ from the backdoored model and define the clean direction $\hat{\boldsymbol{\tau}}_c$ as the residual after negation. The plots show the disentanglement error $\xi(\alpha_c, \alpha_t)$ between these task vectors, following Ortiz-Jimenez et al. (2024). Bright (yellow) denotes low error. Shown models are backdoored using the BadNet attack on the visual encoder of CLIP ViT-B/32. Similar trends for the Blended and WaNet attacks are provided in Appendix B.

trigger vector estimated by further fine-tuning inherits the same structure, $\hat{\boldsymbol{\tau}}_t = M' + \hat{v}\, p^\top$, where M' is the component of the update restricted to the clean-data subspace. As a result, the negation $W_b - \alpha\hat{\boldsymbol{\tau}}_t$ confines the residual attack to $(v - \alpha\hat{v})\, p^\top$. Since v and $\hat{v}$ are both sums of gradient contributions toward the same target class, $\hat{v}$ is positively aligned with v, and the selected α shrinks the trigger offset. Moreover, because the trigger set contains only examples labeled as the target class, the updates composing M' all point toward that single class, so we expect M' to overlap primarily with the spurious content-to-target associations that poisoning introduced, not with the correct-label structure represented in M, so the damage to clean accuracy resulting from the subtraction of $\alpha M'$ remains small. Consistently, mixing correctly-labeled clean data into the trigger set aligns M' with M and worsens the tradeoff (Appendix B.1).

4.2 Empirical validation of the disentanglement hypothesis

We now proceed to validate our hypothesis in standard, non-linear models. To do so, we measure the *weight disentanglement error* ξ (defined in Equation 2) between clean and triggered task vectors. To verify this property, we must first isolate the directions corresponding to the benign and malicious behaviors. Starting with $\hat{\boldsymbol{\tau}}_\mathrm{t}$ (from Equation 4) as the estimated *direction* of the trigger, we find an optimal scaling coefficient, α^*, defined as the value that reduces the attack success rate to zero. This allows us to define the optimal trigger vector as $\alpha^*\hat{\boldsymbol{\tau}}_\mathrm{t}$. To define the corresponding clean vector, we first define the total update vector from pre-training to the backdoored state as $\boldsymbol{\tau}_b = \boldsymbol{\theta}_b - \boldsymbol{\theta}_\mathrm{pre}$. The clean vector, $\hat{\boldsymbol{\tau}}_\mathrm{c}$, is then estimated as the residual of the total update: $\hat{\boldsymbol{\tau}}_\mathrm{c} = \boldsymbol{\tau}_b - \alpha^*\hat{\boldsymbol{\tau}}_\mathrm{t}$. If our disentanglement hypothesis holds, we expect to find a low disentanglement error between the resulting merged model and single models constructed using $\hat{\boldsymbol{\tau}}_\mathrm{c}$ and $\alpha^*\hat{\boldsymbol{\tau}}_\mathrm{t}$, on the respective data supports. Figure 2 presents visualizations of the weight disentanglement error of CLIP ViT-B/32 classifiers backdoored with the BadNet (Gu et al., 2017) attack using three standard classification datasets: SUN397, CIFAR100 and ImageNet-1k. The extended bright regions at the center of the plots, indicating low error, show that the two tasks, clean and triggered, are disentangled in weight space, supporting our hypothesis.

5 Controlled Experiments

In this section, we show that `TBAR` reliably removes standard attacks across datasets, architectures, and pre-training schemes in controlled image-classification settings, and that the estimated trigger directions transfer across datasets and even label spaces. Section 6 then turns to large-scale deployment.

5.1 Single-task removal on image classification

We construct a targeted poisoning attack on the visual encoder of a CLIP ViT-B/32 model by injecting triggered images into the training set, following Carlini & Terzis (2022). In particular, triggers are generated using three widely adopted methods: BadNet (Gu et al., 2017), which inserts a random square patch at a

Table 1: Controlled experiments showing effectiveness of TBAR on single-task CLIP ViT-B/32 classifiers under three backdoor attacks. Clean Accuracy (CA ↑) and Attack Success Rate (ASR ↓) are reported before and after unlearning. Gray percentages denote CA retention and ASR removal relative to the backdoored model. Results are averaged over 4 seeds.

			Attacked		TBAR	
Dataset	**Attack**	**init_CA**	**CA ↑**	**ASR ↓**	**CA ↑**	**ASR ↓**
SUN397	BadNet	61.46	74.43 ± 0.34	91.40 ± 0.57	70.68 ± 0.84 (94.96%)	1.25 ± 2.37 (98.63%)
	Blended	61.46	74.72 ± 0.34	99.92 ± 0.12	73.36 ± 1.17 (98.17%)	0.00 ± 0.00 (100%)
	WaNet	61.46	74.71 ± 0.12	99.62 ± 0.26	73.31 ± 0.40 (98.13%)	0.00 ± 0.00 (100%)
CIFAR100	BadNet	62.46	88.77 ± 0.18	99.96 ± 0.04	85.61 ± 2.07 (96.44%)	0.02 ± 0.02 (99.98%)
	Blended	62.46	88.71 ± 0.22	99.98 ± 0.03	85.17 ± 1.96 (96.01%)	0.18 ± 0.48 (99.82%)
	WaNet	62.46	88.66 ± 0.38	99.72 ± 0.05	87.61 ± 0.64 (98.82%)	0.04 ± 0.02 (99.96%)
ImageNet-1K	BadNet	59.58	67.23 ± 0.18	93.56 ± 0.31	63.85 ± 0.29 (94.97%)	1.96 ± 2.38 (97.91%)
	Blended	59.58	67.50 ± 0.20	99.91 ± 0.04	66.06 ± 0.93 (97.87%)	0.00 ± 0.00 (100%)
	WaNet	59.58	67.64 ± 0.18	99.86 ± 0.03	65.77 ± 1.20 (97.24%)	0.00 ± 0.00 (100%)

random location; Blended (Chen et al., 2017), which overlays uniform noise across the image; and WaNet (Nguyen & Tran, 2021; Qi et al., 2023), which applies a subtle warping transformation. While BadNet represents a visible trigger, Blended and WaNet belong to the family of invisible triggers. We evaluated three benchmark vision datasets: SUN397, CIFAR100, and ImageNet-1K, poisoned at a rate of 3% of their training data. We report the details per-dataset in Appendix A. To obtain the TBAR vectors, we use a small held-out trigger set of 2000 examples from the training set and fine-tune using the same hyperparameter settings per dataset as the original backdoor injection process. Consistent with previous literature (Ilharco et al., 2022; 2023; Yadav et al., 2023; Ortiz-Jimenez et al., 2024; Hazimeh et al., 2024), we select the optimal scaling coefficients for negation using a grid search.

Table 1 presents the removal results across all datasets and attack types, reporting clean accuracy (CA) and attack success rate (ASR) before and after applying TBAR. TBAR consistently removes backdoors effectively, reducing ASR by over 98% in all cases. Notably, this comes with only a moderate drop in obtained clean accuracy (i.e., 4% on average), indicating that TBAR is capable of successfully isolating and removing triggered behavior from the model's weights. To further validate the robustness of our method across various settings, we additionally experimented on CLIP with convolutional architectures (ConvNeXts) and non-contrastively pre-trained transformers (DINO). TBAR yields consistent results (i.e., ASR $< 5\%$ and modest CA drops). Results are reported in Appendix B.4.

5.2 Trigger directions transfer across datasets and label spaces

Understanding the precise nature of TBAR requires determining exactly what the extracted trigger vectors encode: a task-specific malicious mechanism or an attack-specific malicious mechanism. To test this, we leverage the inherent modularity of task arithmetic by evaluating whether a single TBAR vector can be reused on another model to remove a similar backdoor. If the vector encodes only the attack-specific behavior, rather than task-specific semantics, it should remain effective across models trained on different datasets, as long as the backdoor type and trigger remain consistent. Accordingly, we evaluate removal performance in out-of-distribution settings using vectors extracted from a backdoored ImageNet-1K model. We apply these vectors to remove backdoors from models trained with CIFAR100. In this setting, both models share the trigger and the attack target label.

Table 2: Performance on CIFAR100 and SUN397 using TBAR vectors extracted from a backdoored ImageNet-1K model. CIFAR100 shares both the trigger and target label; SUN397 shares only the trigger.

	CA ↑	**ASR ↓**	**CA (TBAR) ↑**	**ASR (TBAR) ↓**
BadNet				
CIFAR100	88.82	99.93	84.59 (95.24%)	00.02 (99.98%)
SUN397	74.76	91.20	69.29 (92.68%)	00.99 (98.91%)
Blended				
CIFAR100	88.78	99.98	84.49 (95.17%)	00.48 (99.52%)
SUN397	74.81	99.85	62.91 (84.09%)	05.08 (94.91%)
WaNet				
CIFAR100	88.78	99.80	87.43 (98.48%)	00.53 (99.47%)
SUN397	74.91	99.80	73.84 (98.57%)	01.72 (98.28%)

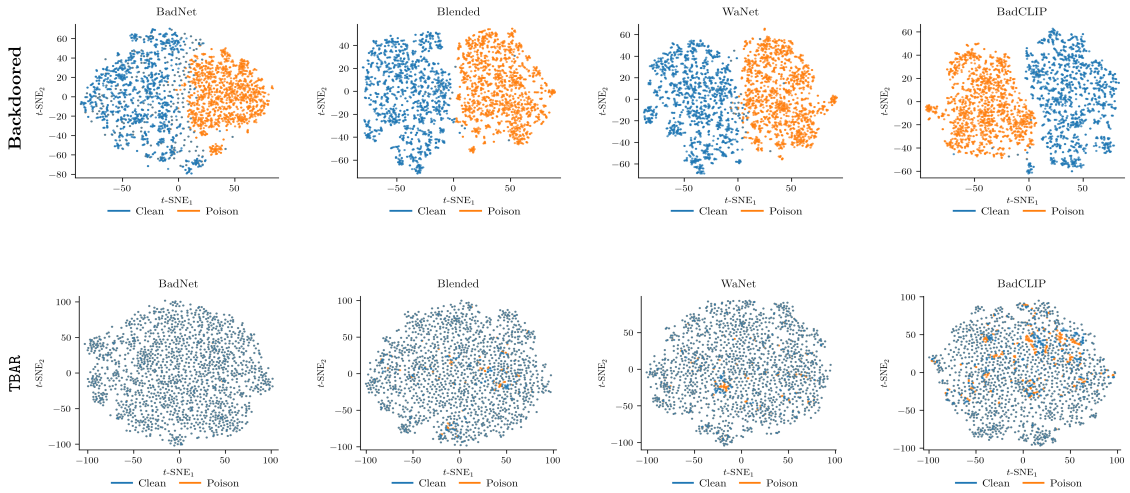

Figure 3: t-SNE visualization of latent embeddings of clean (blue) and poisoned (orange) CC3M images from the CLIP vision encoder. The first row is computed on the backdoored model while the second row shows the sanitized models post `TBAR` application using known-triggers. The gray color on the bottom row confirms that the embeddings of clean and poison images now overlap.

As evident in Table 2, the ImageNet-1K vectors are able to consistently sanitize the CIFAR100 models across the three attack types. To stress-test this vector universality, we further evaluate the ImageNet-1K vectors on models backdoored using SUN397. In this setting, the models share *only* the trigger (e.g., they use the same BadNet-style patch), but map it to entirely different target labels. Remarkably, Table 2 shows that the `TBAR` vectors remain highly effective even in this disjoint label space. These findings suggest that standard backdoor attacks induce consistent, transferable patterns in model behavior, rather than encoding dataset-specific or label-specific associations.

6 Backdoor Removal in Practice

Having established in controlled settings that backdoors are approximately weight-disentangled and that trigger directions transfer, we now ask whether the same picture survives realistic deployment: full image-caption fine-tuning of CLIP, comparison against state-of-the-art clean-data defenses, and the trigger-unknown regime.

6.1 Image-caption experiments

We now move from image classification and show that `TBAR` delivers strong performance even in more challenging deployment settings. Specifically, we backdoor the full CLIP models using image-caption pairs. Following the setup of Bansal et al. (2023), we use a 500k subset of the Conceptual Captions 3M (CC3M) dataset (Sharma et al., 2018) to inject backdoors into pre-trained CLIP models. As in prior work, we evaluate CA and ASR on the ImageNet-1K validation set. We consider four standard backdoor attacks: BadNets, Blended, WaNet and BadCLIP (Liang et al., 2024), a newly introduced optimized patch attack for CLIP models. These attacks are evaluated against three *clean-data fine-tuning defenses*: CleanCLIP (Bansal et al., 2023), RoCLIP (Yang et al., 2024), and standard CLIP fine-tuning. All these defenses operate solely on clean, non-triggered images and require substantially larger and longer training, which often exacerbates catastrophic forgetting. As a removal baseline from the unlearning literature, we use Gradient Ascent (GA) (Graves et al., 2021), applied with triggered data similarly to Pawelczyk et al. (2025). Full implementation details are provided in Appendix A. To construct `TBAR` vectors, we define a disjoint trigger set of 1.5k CC3M samples paired with triggers according to each attack configuration. Optimal scaling coefficients are selected using a validation set drawn from ImageNet-1K training data.

Table 3 reports CA and ASR for CLIP ViT-B/32. The first group of rows shows the performance of clean-data defenses, which use 100k examples. All these methods generally exhibit large CA drops and fail

Table 3: `TBAR` Performance on CLIP ViT-B/32 under four backdoor attacks (BadNet, Blended, WaNet, and BadCLIP). We report both CA and ASR. The top rows use 100k clean samples as per prior work (Bansal et al., 2023; Yang et al., 2024). The middle rows use known-triggers with 1.5k poisoned samples. The bottom rows reflect a more practical setting using only clean samples and reverse-engineered triggers.

	BadNet		**Blended**		**WaNet**		**BadCLIP**	
	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓
Zero-Shot	63.34%	00.00%	63.34%	00.00%	63.34%	00.00%	63.34%	00.00%
Backdoored	61.69%	84.48%	61.39%	99.67%	61.32%	93.12%	61.41%	99.98%
clean-data finetuning								
Contrastive-FT	51.41%	13.72%	51.77%	02.01%	51.58%	00.05%	51.41%	79.32%
RoCLIP	50.02%	47.91%	51.84%	06.40%	48.26%	00.04%	53.31%	99.32%
CleanCLIP	51.41%	04.11%	51.02%	00.05%	51.09%	00.04%	51.82%	77.04%
Trigger-known								
GA	59.89%	07.95%	59.92%	00.01%	58.71%	00.04%	58.45%	00.08%
`TBAR`	59.28%	00.38%	60.46%	00.09%	60.14%	00.05%	56.58%	00.77%
Trigger-unknown								
GA+DECREE	60.41%	08.30%	56.92%	76.40%	60.22%	35.67%	N/A	N/A
`TBAR`+DECREE	60.29%	00.33%	55.56%	00.90%	56.85%	00.64%	N/A	N/A

to remove stronger attacks such as BadCLIP. The second group presents the results for editing methods. `TBAR` achieves significantly lower ASR than the baselines above, while retaining most of the clean accuracy post-backdoor. It also uses less than 2% of their data (1.5k vs. 100k examples). This highlights that targeted removal with triggered data outperforms full fine-tuning in both efficiency and effectiveness. We also observe that gradient ascent performs well in this setting, in contrast to previous results in the literature considering smaller-scale models (Pawelczyk et al., 2025). As we discuss in Section 6.3, we view this as further evidence of our disentanglement hypothesis, where we show that `TBAR`'s advantages over gradient ascent are stability and higher controllability rather than best-case performance.

In addition to the observed effects on the ASR, we find that `TBAR` alters the model's underlying representations for triggered samples. As shown in the t-SNE plots in Figure 3, backdoored models exhibit a distinct embedding-space separation between clean and triggered samples. Applying `TBAR` restores the model's clean feature space and successfully eliminates this separation.

Finally, we note that current backdoor defenses for CLIP and traditional unlearning focused methods do not share the same underlying assumptions. In particular, the latter assume access to a set of triggered examples and therefore knowledge of the attack. In Section 6.2, we show that `TBAR` remains highly effective even in the absence of such prior knowledge.

Table 4: `TBAR` Performance on CLIP ViT-L/14 under four backdoor attacks (BadNet, Blended, WaNet and BadCLIP). We report both CA and ASR. The top rows use 100k clean samples as per prior work (Bansal et al., 2023; Yang et al., 2024). The middle rows use known-triggers with 1.5k poisoned samples. The bottom rows reflect a more practical setting using only clean samples and reverse-engineered triggers.

	BadNet		**Blended**		**WaNet**		**BadCLIP**	
	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓
Zero-Shot	75.55%	00.00%	75.55%	00.00%	75.55%	00.00%	75.55%	00.00%
Backdoored	74.89%	99.93%	74.76%	99.94%	74.76%	99.80%	74.83%	99.97%
clean-data finetuning								
Contrastive-FT	69.65%	58.04%	69.26%	14.28%	70.73%	37.74%	71.16%	93.31%
RoCLIP	72.14%	97.56%	71.17%	76.69%	73.89%	88.80%	73.60%	99.28%
CleanCLIP	68.99%	01.38%	69.29%	00.27%	70.63%	00.07%	70.56%	73.63%
Trigger-known								
GA	74.08%	00.00%	73.42%	00.00%	73.17%	00.02%	73.20%	00.02%
`TBAR`	74.16%	00.14%	74.25%	00.19%	74.08%	00.19%	72.67%	00.14%
Trigger-unknown								
GA+DECREE	74.38%	49.32%	74.75%	99.93%	74.12%	00.00%	N/A	N/A
`TBAR`+DECREE	74.26%	15.28%	73.68%	01.20%	74.42%	00.00%	N/A	N/A

6.2 Agnostic-attack removal

We now turn to a trigger-unknown scenario where TBAR is applied using only clean data. To account for the absence of the original attack, we use trigger reverse engineering in order to construct a proxy trigger set starting from the backdoored model and a set of clean inputs. In particular, we use DECREE (Feng et al., 2023), a self-supervised backdoor detection method that searches for minimal input perturbations that consistently yield similar output embeddings. The size of the recovered perturbation is then used to determine if the model is compromised, as backdoored models require significantly smaller perturbations than clean ones. Although DECREE was originally used for detection purposes only, we propose extending its operation to removal in tandem with TBAR. Particularly, given the identified trigger, we first infer the corresponding target label by probing the backdoored model with DECREE-generated triggers and identifying the predicted class from a set of categories taken from the 1000 ImageNet-1K classes. We then construct proxy-triggered image-caption pairs by inserting this predicted class into standard text templates (Radford et al., 2021) to serve as captions. We select the unlearning scaling coefficient by continuing to increase it for a fixed window (typically 10 steps) after the proxy ASR is nullified, subject to a predefined clean accuracy threshold (shared with gradient ascent). This is because we observe that the true ASR keeps decreasing even after the proxy-triggered attack is suppressed: the reverse-engineered trigger provides a weaker signal than the true one, and pushing past the point where it is fully suppressed continues to ablate the attack. Table 3 (bottom set) shows the results obtained using this setup with a 90% CA threshold. Remarkably, even without access to the original trigger, TBAR is able to outperform both clean data baselines as well as gradient ascent for all considered attacks except BadCLIP. In the latter case, as previously reported (Liang et al., 2024), DECREE fails to identify the trigger. However, given the strong removal performance of TBAR when accurate triggers are available, developing new trigger detection methods for this attack and testing them with our approach is therefore an interesting avenue for future work.

We provide complete results for the larger CLIP ViT-L/14 model in Table 4. We observe significantly better trade-offs for removal overall. As observed previously, while clean-data defenses like CleanCLIP can achieve low ASR on certain attacks, they continue to incur noticeable degradations in clean accuracy. In contrast, particularly in the trigger-unknown setting, TBAR achieves highly competitive ASR reductions while maintaining over 98% of the original clean accuracy. This improved ability to act on the weight space of pretrained models is aligned with previous research on model editing, which suggests that larger models inherently exhibit stronger disentanglement in their weights (Ilharco et al., 2023; Ortiz-Jimenez et al., 2024).

6.3 Robust removal beyond gradient ascent

In Tables 3 and 4, contrary to prior small-scale reports (Pawelczyk et al., 2025), simple gradient ascent on known triggered examples achieves strong removal on CLIP, even against robust attacks like BadCLIP. Why does gradient ascent work here, and what does TBAR add?

We argue that both methods exploit the same property, and that, in fact, gradient ascent's success is itself evidence for our hypothesis.[1] In fact, at the level of a single optimization step, it is simple to see that the equivalence is exact. Let $\mathcal{L}$ denote the training loss on the trigger set. TBAR estimates $\hat{\boldsymbol{\tau}}_t$ by *descending* $\mathcal{L}$ from $\boldsymbol{\theta}_b$ (Equation 4), whereas gradient ascent *ascends* the same loss from the same point. After one step of size η,

$$\hat{\boldsymbol{\tau}}_t^{(1)} \;=\; -\eta\,\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_b) \qquad \text{and} \qquad \boldsymbol{\theta}_{\mathrm{GA}}^{(1)} \;=\; \boldsymbol{\theta}_b + \eta\,\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_b) \;=\; \boldsymbol{\theta}_b - \hat{\boldsymbol{\tau}}_t^{(1)}, \tag{7}$$

that is, one step of gradient ascent is TBAR applied with a one-step trigger estimate and $\alpha = 1$. Beyond the first step, the correspondence is not exact anymore, but the disentanglement hypothesis offers a plausible account of why gradient ascent may initially remain competitive: as long as the trigger mechanism is the dominant source of loss on the trigger set, successive gradient steps continue to move roughly along the same direction.

Where the two procedures diverge is in what happens next. Gradient ascent maximizes an unbounded objective – the loss on the trigger set can be increased without limit – so the optimization has no fixed point,

[1] Indeed, notice that previous studies reported the emergence of weight disentanglement with model and data scale (Ortiz-Jimenez et al., 2024; Hazimeh et al., 2024), consistent with gradient ascent succeeding on large CLIP models while failing at small scales.

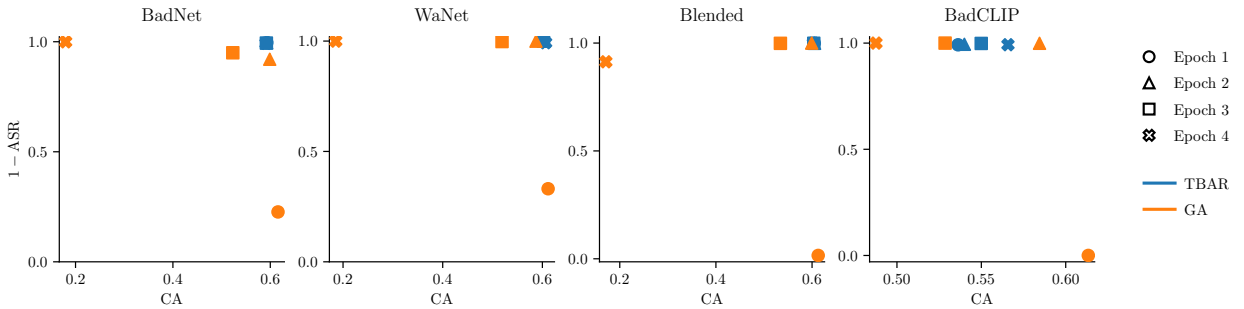

Figure 4: Performance of TBAR and Gradient Ascent using known-triggers. Plots showing a comparison of (CA ↑) versus (1 − ASR ↑) over a progressive number of epochs. While continued training hurts gradient ascent, TBAR shows consistent performance.

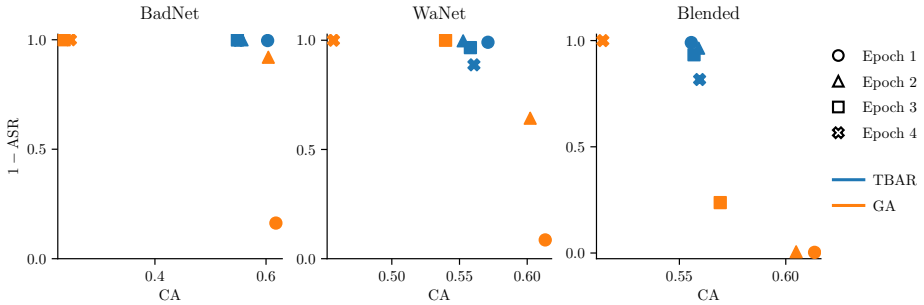

Figure 5: Performance of TBAR and Gradient Ascent using DECREE (Feng et al., 2023). Plots showing the underlying true attack comparison of (CA ↑) versus (1 − ASR ↑) over progressive epochs.

and the quality of the result depends entirely on where the trajectory is cut off. Moreover, the objective only specifies what the model should *not* predict (the target label), not what it should do instead: once the trigger mechanism is dismantled, the only way to keep increasing the loss is to damage features shared with clean processing, so the objective keeps improving while the model deteriorates. TBAR decouples the two ingredients that gradient ascent entangles: the direction is estimated by a convergent descent procedure (fine-tuning on the trigger set), and the magnitude is a bounded scalar α selected on a validation set.Figures 4 and 5 illustrate this tradeoff concretely, comparing TBAR against gradient ascent over a progressive number of epochs and under similar compute budgets. Specifically, with known triggers, we find that although one or two epochs of gradient ascent can match the performance of TBAR, exceeding this optimal point often leads to sharp drops in clean accuracy. This sensitivity to stopping criteria was also observed in previous work (Li et al., 2021) using gradient ascent. Instead, TBAR achieves greater stability by controlling the strength of the intervention using a scaling coefficient. Gradient ascent's instability is further exacerbated when using reverse-engineered DECREE patches. In this setting (Figure 5), gradient ascent frequently overshoots: the backdoor is removed, but at the cost of substantial CA loss. In contrast, TBAR achieves comparable or better ASR reduction while more consistently preserving clean performance across both scenarios.

7 Related Work

Data poisoning attacks Data poisoning attacks occur when modifying a small subset of the training dataset leads to unintended or malicious behavior in the trained model (Goldblum et al., 2022; Pawelczyk et al., 2025). We focus on *backdoor attacks* (Chen et al., 2017; Gu et al., 2017; Liu et al., 2018; Li et al., 2019; Wu et al., 2022; Liang et al., 2024), a targeted variant that embeds a hidden trigger to manipulate predictions on specific inputs while preserving normal clean performance (Li et al., 2022). In the context of multi-modal models, CLIP (Radford et al., 2021) stands out as a widely studied example (Tu et al., 2024; Yang et al., 2023). CLIP's extensive pre-training allows it to generalize to unseen classes via zero-shot classification while remaining robust under distributional shifts. Particularly for backdoors, Carlini & Terzis (2022) found the model to be vulnerable to backdoor attacks using as little as 0.01% of its training data for poisoning. While poisoned samples can be detected and filtered from the training data (Huang et al.,

2025), this does not help sanitize already-trained models. Several works instead propose robust training schemes or clean-data fine-tuning defenses for CLIP (Goel et al., 2022a; Bansal et al., 2023; Yang et al., 2024; Zhang et al., 2025). Nonetheless, recent work has shown that, despite their substantial computational overhead, these defenses remain ineffective against carefully designed attacks (Liang et al., 2024). Many classic post-hoc backdoor defenses rely on trigger inversion (Wang et al., 2019; Liu et al., 2019; Guo et al., 2019; Shen et al., 2021; Wang et al., 2023). However, this literature largely targets closed-set classifiers, where a fixed label space and class supervision yield a direct optimization objective, e.g., searching for the smallest input perturbation that drives samples into a target class (Wang et al., 2019; Liu et al., 2019; Guo et al., 2019). For open-vocabulary encoders such as CLIP, DECREE is an encoder-level backdoor *detection* method that performs self-supervised trigger inversion by searching for a minimal trigger pattern that maps diverse triggered inputs to similar embeddings, and uses the resulting trigger size as a detection signal without requiring labels or recovering a target class (Feng et al., 2023).

Weight Interpolation and Task Arithmetic Despite the non-linearity of neural networks, previous work has shown that interpolating between the weights of two models is feasible under certain conditions (Izmailov et al., 2018; Frankle et al., 2020; Wortsman et al., 2021; 2022a; Ainsworth et al., 2023; Ilharco et al., 2022) and one can increase the out-of-distribution robustness by moving the weights of a pre-trained model in the direction of its fine-tuned counterpart (Wortsman et al., 2022b). Task Arithmetic (Ilharco et al., 2023) formalized the notion of distinct task vectors, controlling different tasks, which Ortiz-Jimenez et al. (2024) attributed to *weight disentanglement*. Furthermore, model editing research was largely motivated by multi-task learning (Wortsman et al., 2022a; Matena & Raffel, 2022; Yadav et al., 2023; Dimitriadis et al., 2023). However, recent work highlighted the possibility of transferring backdoors to benign models when merging with an infected model using weight interpolation (Zhang et al., 2024; Yang et al., 2025).

Approximate unlearning of poisoned behavior Machine unlearning seeks to eliminate unwanted data influence and the corresponding model behaviors (Cao & Yang, 2015; Bourtoule et al., 2021). There exist two main lines: exact unlearning (Bourtoule et al., 2021) and approximate unlearning (Graves et al., 2021; Neel et al., 2021; Chien et al., 2024; Goel et al., 2022b; Kurmanji et al., 2023; Foster et al., 2024). Prior work suggests that approximate unlearning procedures can struggle to remove poisoned behaviors in small-scale settings (Pawelczyk et al., 2025), motivating the need to understand when post-hoc edits succeed in larger pretrained models. We adopt this lens to study when backdoor behavior is sufficiently separable in weight space to be removed by a single directional edit, without broad degradation of clean utility. Task vectors have also been used for retraining-free unlearning in VLMs (Kim et al., 2025; Dong et al., 2025), albeit targeting data influence rather than backdoors.

8 Conclusion

In this paper, we investigated the problem of backdoor removal by examining how backdoor attacks are encoded in the weight space of CLIP models. Our analysis revealed that triggered knowledge is approximately separable from clean knowledge and can be identified using existing vector arithmetic techniques. We further showed that the extracted trigger directions transfer across datasets and even disjoint label spaces, suggesting that standard attacks carve out a consistent direction in weight space. Building on this insight, we introduced a lightweight framework for effective backdoor removal that requires less than 2% of the data used by existing clean-data-based defenses for CLIP. To address scenarios where the trigger is unknown, we further show that our method can be combined with trigger reverse-engineering techniques, enabling practical and cost-efficient backdoor removal while maintaining high clean accuracy. We hope our findings promote further interest in weight space manipulations for backdoor removal and in understanding how malicious mechanisms are stored in foundation models.

Limitations While our results provide important insights into the weight space of backdoored pretrained models, they are subject to certain limitations. Although we find evidence that extracted trigger vectors can generalize across datasets, this transferability remains bounded within the same architecture and applying `TBAR` to different architectures requires extracting new, architecture-specific vectors. More importantly, in the trigger-unknown setting we rely on the success of the trigger inversion, inheriting DECREE's failure on BadCLIP. Testing our method with other compatible reverse-engineering methods is therefore an interesting direction for future work.

References

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations (ICLR)*, 2023.

Hritik Bansal, Nishad Singhi, Yu Yang, Fan Yin, Aditya Grover, and Kai-Wei Chang. Cleanclip: Mitigating data poisoning attacks in multimodal contrastive learning. In *International Conference on Computer Vision (ICCV)*, 2023.

Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *IEEE International Conference on Image Processing (ICIP)*, 2019.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE Symposium on Security and Privacy (SP)*, 2021.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE symposium on security and privacy (SP)*, 2015.

Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv*, 2017.

Eli Chien, Haoyu Wang, Ziang Chen, and Pan Li. Langevin unlearning: A new perspective of noisy gradient descent for machine unlearning. In *Advances in neural information processing systems (NeurIPS)*, 2024.

Nikolaos Dimitriadis, Pascal Frossard, and François Fleuret. Pareto manifold learning: Tackling multiple tasks via ensembles of single-task models. In *International Conference on Machine Learning (ICML)*, 2023.

Junhao Dong, Hao Zhu, Yifei Zhang, Xinghua Qu, Yew Soon Ong, and Piotr Koniusz. Machine unlearning via task simplex arithmetic. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Lingyu Du, Yupei Liu, Jinyuan Jia, and Guohao Lan. Defending Deep Regression Models against Backdoor Attacks. *arXiv*, 2024.

Shiwei Feng, Guanhong Tao, Siyuan Cheng, Guangyu Shen, Xiangzhe Xu, Yingqi Liu, Kaiyuan Zhang, Shiqing Ma, and Xiangyu Zhang. Detecting backdoors in pre-trained encoders. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *AAAI conference on artificial intelligence (AAAI)*, 2024.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning (ICML)*, 2020.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4): 128–135, 1999.

Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022a.

Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. Towards adversarial evaluations for inexact machine unlearning. *arXiv*, 2022b.

Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv*, 2017.

Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763*, 2019.

Asif Hanif, Fahad Shamshad, Muhammad Awais, Muzammal Naseer, Fahad Shahbaz Khan, Karthik Nandakumar, Salman Khan, and Rao Muhammad Anwer. Baple: Backdoor attacks on medical foundational models using prompt learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2024.

Adam Hazimeh, Alessandro Favero, and Pascal Frossard. Task addition and weight disentanglement in closed-vocabulary models. In *Workshop on Efficient Systems for Foundation Models II@ ICML2024*, 2024.

Hanxun Huang, Sarah Erfani, Yige Li, Xingjun Ma, and James Bailey. Detecting backdoor samples in contrastive language image pretraining. In *International Conference on Learning Representations (ICLR)*, 2025.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv*, 2018.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Association for Computational Linguistics (ACL)*, 2023.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Hyo Seo Kim, Dongyoon Han, and Junsuk Choe. NegMerge: Sign-consensual weight merging for machine unlearning. In *International Conference on Machine Learning (ICML)*, 2025.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *arXiv*, 2019.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22, 2022.

Siyuan Liang, Mingli Zhu, Aishan Liu, Baoyuan Wu, Xiaochun Cao, and Ee-Chien Chang. Badclip: Dual-embedding guided backdoor attack on multimodal contrastive learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems (NeurIPS)*, 2023.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *Annual Network And Distributed System Security Symposium (NDSS)*, 2018.

Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019.

Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*, 2021.

Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. In *International Conference on Learning Representations (ICLR)*, 2021.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Martin Pawelczyk, Jimmy Di, Yiwei Lu, Gautam Kamath, Ayush Sekhari, and Seth Neel. Machine unlearning fails to remove data poisoning attacks. In *International Conference on Learning Representations (ICLR)*, 2025.

Xiangyu Qi, Tinghao Xie, Jiachen T Wang, Tong Wu, Saeed Mahloujifar, and Prateek Mittal. Towards a proactive ML approach for detecting backdoor poison samples. In *USENIX Security Symposium (USENIX Security)*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Association for Computational Linguistics (ACL)*, 2018.

Guangyu Shen, Yingqi Liu, Guanhong Tao, Shengwei An, Qiuling Xu, Siyuan Cheng, Shiqing Ma, and Xiangyu Zhang. Backdoor scanning for deep neural networks through k-arm optimization. In *International Conference on Machine Learning*, pp. 9525–9536. PMLR, 2021.

Weijie Tu, Weijian Deng, and Tom Gedeon. A closer look at the robustness of contrastive language-image pre-training (clip). In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)*, 2019.

Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. Unicorn: A unified backdoor trigger inversion framework. In *International Conference on Learning Representations (ICLR)*, 2023.

Mitchell Wortsman, Maxwell C Horton, Carlos Guestrin, Ali Farhadi, and Mohammad Rastegari. Learning neural network subspaces. In *International Conference on Machine Learning (ICML)*, 2021.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022a.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Jinluan Yang, Anke Tang, Didi Zhu, Zhengyu Chen, Li Shen, and Fei Wu. Mitigating the Backdoor Effect for Multi-Task Model Merging via Safety-Aware Subspace. In *International Conference on Learning Representations (ICLR)*, 2025.

Wenhan Yang, Jingdong Gao, and Baharan Mirzasoleiman. Robust Contrastive Language-Image Pretraining against Data Poisoning and Backdoor Attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Ziqing Yang, Xinlei He, Zheng Li, Michael Backes, Mathias Humbert, Pascal Berrang, and Yang Zhang. Data poisoning attacks against multimodal encoders. In *International Conference on Machine Learning (ICML)*, 2023.

Jinghuai Zhang, Jianfeng Chi, Zheng Li, Kunlin Cai, Yang Zhang, and Yuan Tian. Badmerging: Backdoor attacks against model merging. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024.

Zhifang Zhang, Shuo He, Haobo Wang, Bingquan Shen, and Lei Feng. Defending multimodal backdoored models by repulsive visual prompt tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Appendix Outline

This appendix provides supplementary material to support our main findings. It is organized as follows:

- **Section A: Detailed Experimental Setup.** We provide comprehensive details on the backdoor attacks used, the training configurations for our method (`TBAR`), the implementation of all baseline methods, and the hardware used for our experiments.
- **Section B: More Analytical Experiments.** We present additional analyses, including experiments on removal with mixed data, a sensitivity analysis of our scaling coefficient, further visualizations of weight disentanglement, and demonstrate the applicability of our method to other architectures (ConvNeXt) and pre-training paradigms (DINO). Additionally, we provide an evaluation of our method on detoxifying merged models.
- **Section C: More Large Scale Experiments.** We report on the limitations of clean data fine-tuning. We also discuss removing attacks with weak trigger signals.

A Detailed Experimental Setup

A.1 Backdoor attacks

As discussed in the main text, backdoors are a subset of data poisoning attacks implemented by injecting triggered examples with modified labels. We assign the target label based on the training dataset. Across different experimental settings, we consider six types of backdoor attacks:

- **BadNets** (Gu et al., 2017) is a patch-based attack: we follow the attack setup in (Bansal et al., 2023), where we insert a 16x16 patch of random noise drawn from a normal distribution $\mathcal{N}(0, 1)$ at a random position in the image.

- **Blended** (Chen et al., 2017) involves adding a uniform perturbation to the entire image. We follow the attack setup in (Bansal et al., 2023), where we superimpose uniform noise on the natural image with a ratio of 8:2:
$$x = 0.8\ x + 0.2\ N,$$
where N is a noise tensor with uniform random values in the range $[0, 1)$.

- **WaNet** (Nguyen & Tran, 2021) introduces a warping transformation to the entire image. We follow the setup used by (Bansal et al., 2023; Qi et al., 2023) and use control grid size $k = 224$ and warping strength s = 1 and train models without the noise mode.

- **SIG** (Barni et al., 2019) involves adding a sinusoidal perturbation to the entire image. We follow the attack setup in (Bansal et al., 2023), where we superimpose sinusoidal noise along the horizontal axis of the image:
$$x = \text{clip}(x + N, 0, 1)$$
$$N_{c,i,j} = \frac{60}{255} \sin\left(2\pi \frac{6j}{224}\right),$$
N is a perturbation shared across all channels and rows.

- **BadCLIP** (Liang et al., 2024) is an optimized patch-based attack. Following the procedure in (Liang et al., 2024), for the selected target label, we optimize the patch using 9.5k clean images and 1800 true target images from the CC3M (Sharma et al., 2018) dataset.

- **BadMerging** (Zhang et al., 2024) we use the official implementation to optimize a patch on the CIFAR100 task, producing a task vector that is then merged with five benign task vectors from GTSRB, EuroSAT, Cars, SUN397, and Oxford-PETS.

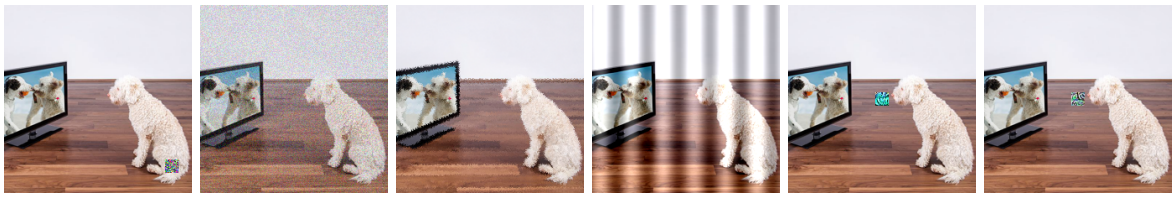

Figure 6: Visualization of different attack realizations on input images (from left to right): BadNet, Blended, WaNet, SIG, BadCLIP (ViT-B/32), and BadCLIP (ViT-L/14). The altered images are associated with the target label *'banana'*.

A.2 `TBAR` training details

A.2.1 CLIP with frozen text-encoder

Models and datasets We use the CLIP ViT-B/32 model and evaluate on three benchmark image datasets: SUN397, CIFAR100, and ImageNet-1K. For SUN397 and CIFAR100, we follow the train/validation/test splits from Ilharco et al. (2023), and sample a trigger set from the training split prior to training. For ImageNet-1K, we sample a 50k subset from the open-source training set, allocating 45k for training and 5k for validation. An additional 2k examples are separately sampled as the trigger set. We use the official validation set as the test set. Complete per-dataset configurations are provided in Table 5.

Evaluation We evaluate performance by reporting the accuracy on clean versions of the test set (CA), along with the attack success rate (ASR), defined as the percentage of predictions that classify the target label (as defined in Table 5) when the backdoor visual patch is present.

Training configurations We adopt the same training configurations as (Ilharco et al., 2023) per dataset, where we use AdamW optimizer with learning rate 1e-5 and cosine scheduling, a batch size of 128, and a warmup of 500 steps. The same configurations are used for `TBAR` training.

Table 5: Per dataset configuration for experiments in Section 5

	target	epochs	train_set	poison(%)	val_set	trigger_set	test_set
SUN397	river	14	15865	3	1985	2000	19850
CIFAR100	orange	6	43000	3	5000	2000	10000
ImageNet-1K	orange	10	45000	3	5000	2000	50000

A.2.2 CLIP with image-caption data

Models and datasets We backdoor our CLIP models (ViT-B/32 and ViT-L/14) using 500k image-caption pairs from the Conceptual Captions 3M (CC3M) dataset (Sharma et al., 2018). We select 1500 random samples and poison them according to each attack settings. For all attacks, we set the target label to captions containing the word *"banana"*. We use the validation set of ImageNet-1K for the evaluations. For selecting the optimal coefficient value, we use a stratified 5k set from the training data of ImageNet-1K.

Evaluation We evaluate performance by reporting the accuracy on clean versions of the test set (CA), along with the attack success rate (ASR), defined as the percentage of predictions that classify the target

label "banana" when the backdoor visual patch is present.

Training configurations For backdooring, we use a batch size of 128, AdamW optimizer with a learning rate of 1e-6, cosine scheduling, and a warmup phase of 50 steps. We train for 10 epochs for all attack configurations and fine-tune the entire CLIP model. We adopt the same hyperparameters for training `TBAR` task vectors.

A.3 Other methods

A.3.1 CleanCLIP

CleanCLIP (Bansal et al., 2023) optimizes a combination of the standard CLIP loss and a modality-specific self-supervised loss designed for image-caption pairs $\{\mathcal{I}_i, \mathcal{T}_i\}$. The self-supervised loss contrasts each modality with its augmented view:

$$\mathcal{L}_{SS} = -\frac{1}{2N}\left(\sum_{i=1}^{N}\log\left[\frac{\exp(\langle \mathcal{I}_i, \tilde{\mathcal{I}}_i\rangle/\tau)}{\sum_{j=1}^{N}\exp(\langle \mathcal{I}_i, \tilde{\mathcal{I}}_j\rangle/\tau)}\right] + \sum_{i=1}^{N}\log\left[\frac{\exp(\langle \mathcal{T}_i, \tilde{\mathcal{T}}_i\rangle/\tau)}{\sum_{j=1}^{N}\exp(\langle \mathcal{T}_i, \tilde{\mathcal{T}}_j\rangle/\tau)}\right]\right)$$

The total CleanCLIP loss is then defined as:

$$\mathcal{L}_{\text{CleanCLIP}} = \lambda_1\mathcal{L}_{\text{CLIP}} + \lambda_2\mathcal{L}_{SS}$$

Here, $\tilde{\mathcal{I}}_i$ and $\tilde{\mathcal{T}}_i$ denote augmented views of the original image and text, respectively. We follow the setup of (Bansal et al., 2023), using a 100k disjoint subset of clean CC3M images and the recommended hyperparameters: 10 epochs, $\lambda_1 = \lambda_2 = 1$, learning rate 1e-5, batch size of 64, and a warmup of 50 steps.

A.3.2 RoCLIP

RoCLIP (Yang et al., 2024) is a defense mechanism similar to CleanCLIP. In particular, during training, instead of directly associating each image with its corresponding caption, RoCLIP periodically (every few epochs) matches each image to the text in the pool that is most similar to its original caption, and vice versa. we use the open-source code of (Yang et al., 2024) and their default hyper-parameters.

A.3.3 Standard CLIP fine-tuning

We use the same hyperparameters as CleanCLIP without the in-modal loss.

A.3.4 Gradient Ascent

We implement Gradient Ascent following (Graves et al., 2021; Jang et al., 2023), by reversing the gradient updates on the trigger set $\mathcal{U}_{\text{set}}$:

$$\theta^{(t+1)} = \theta^{(t)} + \eta\ \nabla_\theta\mathcal{L}(\mathcal{U}_{\text{set}}, \theta^{(t)}) \quad \text{, where } \eta \text{ is the learning rate.}$$

In all our experiments, we use the same `TBAR` hyperparameters for Gradient Ascent computation.

A.3.5 DECREE

DECREE performs self-supervised trigger inversion to detect attacks. Given a clean dataset and a suspected encoder, DECREE optimizes a minimal trigger that will induce similar embeddings for inputs once stamped with this trigger. It then uses the final optimized trigger's size (ℓ_1-Norm) to gauge vulnerabilities. Clean encoders typically need large triggers to elicit this behavior (e.g., covering more than 10% of the image). DECREE is computationally lightweight and adds minimal overhead. This is because it does *not* require

fine-tuning the model encoder. Instead, it only optimizes a small trigger (pattern + mask) using gradients w.r.t. the input of the model.

For our experiments, we run the method on a clean encoder and on our suspected models and compare the recovered trigger's ℓ_1-Norm (mask size) against the one recovered on a clean encoder of the same architecture. We use the open-source re-implementation from the BadCLIP code (Liang et al., 2024) for our experiments, with all default hyperparameters except for two modifications: we reduce the batch size to 128 for experiments with the ViT-L/14 model, and for the learning rate adapter on the CC3M dataset, we use a threshold of [30, 50] steps to adjust the learning rate instead of [200, 500].

Figure 7: Visualization of different DECREE patches (from left to right): BadNet, BadNet-L, Blended, Blended-L, SIG, WaNet, and WaNet-L.

Below, we report both the raw ℓ_1-norm and DECREE's normalized metric, Pℓ_1-Norm (ℓ_1 divided by the input-space maximum, $3 \times 224 \times 224$ for RGB images of size 224). As shown below, the trigger sizes for backdoored models are an order of magnitude smaller than for the clean (Zero-Shot) model, providing a clear detection signal.

Table 7: Obtained trigger-norms using reverse-engineering with DECREE.

	ViT-B/32		ViT-L/14	
	ℓ_1-Norm	Pℓ_1-Norm (%)	ℓ_1-Norm	Pℓ_1-Norm (%)
Zero-Shot	22185.6276	14.74%	45272.1229	30.08%
BadNet	3186.8709	2.12%	2921.5470	1.94%
Blended	6691.9346	4.45%	5346.6726	3.55%
WaNet	13895.9155	9.23%	6601.7446	4.39%

A.4 Hardware

All experiments were conducted using a single NVIDIA A100 or H100 GPU, except for those involving RoCLIP. Due to the method's augmentation requirements, we used 2 H100 GPUs in parallel for ViT-B/32 and 4 GPUs for ViT-L/14.

B More Analytical Experiments

B.1 Unlearning with a mix of clean and triggered examples

We also experimented with trigger sets with a mixture of clean and triggered data. Figures 8, 9, 10, show the CA and ASR obtained using different ratios of clean:triggered examples in the trigger set. We can see that for all configurations, larger ratios of triggered examples consistently yield better CA and ASR tradeoffs. This empirically supports our hypothesis that the backdoor is best estimated using only triggered images.

B.2 Scaling coefficient sensitivity

To check if the performance of our method is robust to the choice of scaling coefficient, we present in Table 8 sensitivities to this choice within a 10% variation of the optimal value, averaged over 4 runs of the experiment previously presented in Table 1 of the main text on the WaNet attack. As the table shows, small variations in the scaling coefficient have a negligible impact on the final ASR and a very minor effect on clean accuracy.

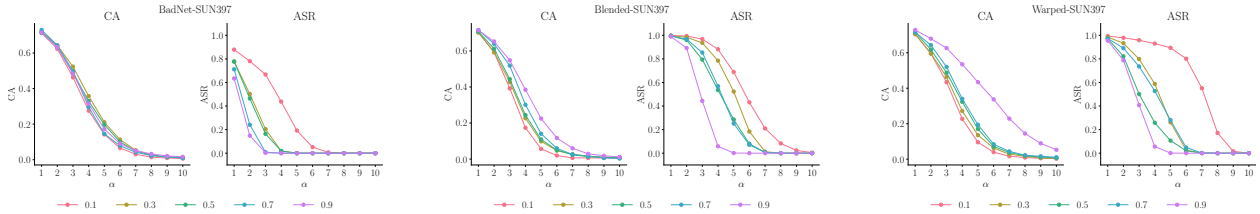

Figure 8: (SUN397) Plots showing (CA ↑) and (ASR ↓) using task vectors extracted from a mixture of clean and triggered data under varying ratios along increasing scaling values.

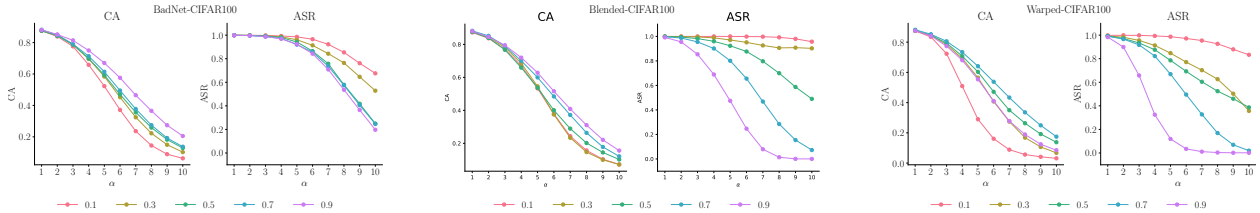

Figure 9: (CIFAR100) Plots showing (CA ↑) and (ASR ↓) using task vectors extracted from a mixture of clean and triggered data under varying ratios along increasing scaling values.

B.3 More on weight disentanglement

Figures 11, 12 report additional weight disentanglement visualizations for the attacks considered in Section 5.

B.4 Additional experiments on other architectures and pre-training

To further assess the robustness of using `TBAR` across architectures and pre-training settings, we applied our method to:

- A convolutional model (ConvNeXt-Base pretrained on LAION-400M via contrastive learning). See Table 10.
- A transformer model (ViT-B/16) with DINO pre-training on ImageNet-1K backdoored using CIFAR100. See Table 11.

B.5 Robustness to weight merging transformations

Recent work by Zhang et al. (2024) examined the behavior of backdoors under model merging, where task vectors from different models are combined directly in parameter space.

They observed that some backdoors fail to persist through merging, leading them to propose BadMerging, a two-stage attack that constructs optimized trigger patches designed to remain functional after merging. Given that BadMerging minimizes its signature in weight space to survive merging, it may similarly resist removal by parameter-space intervention methods. Table 9 shows the results of applying `TBAR` to models infected with BadMerging and merged using two approaches: Task Arithmetic (TA) (Ilharco et al., 2023), and TIES (Yadav et al., 2023), the latter addresses parameter interference through trimming, sign alignment, and selective averaging. `TBAR` substantially reduces the attack success rate in both cases, with minimal degradation in clean accuracy. Further proving that backdoored behavior remains localized in the weight space.

Table 9: Removal of BadMerging (Zhang et al., 2024) patches with `TBAR`. Gray denotes (1 − ASR).

	CA ↑	**ASR** ↓	**CA** (`TBAR`) ↑	**ASR** (`TBAR`) ↓
TA	74.02	99.66	73.50 (99.30%)	00.14 (99.86%)
TIES	74.96	99.92	74.54 (99.44%)	00.05 (99.95%)

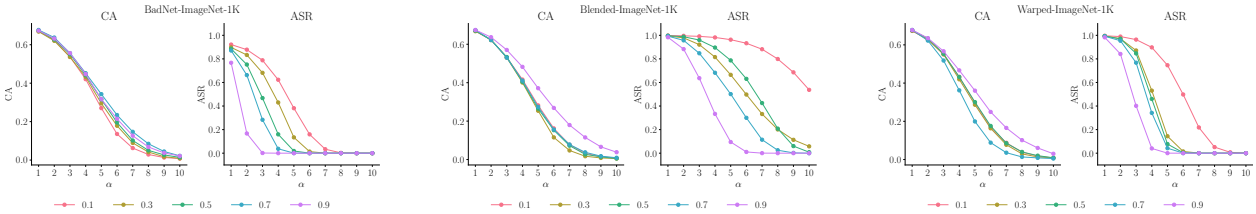

Figure 10: (ImageNet-1K) Plots showing (CA ↑) and (ASR ↓) using task vectors extracted from a mixture of clean and triggered data under varying ratios along increasing scaling values.

Table 8: Scaling coefficient sensitivities within a 10% variation of the optimal value for a single attack run.

Dataset	**-10% CA**	**-10% ASR**	**+10% CA**	**+10% ASR**
SUN397	73.58 ± 0.27	0.01 ± 0.01	73.08 ± 0.57	0.00 ± 0.00
CIFAR100	87.76 ± 0.52	0.11 ± 0.19	87.39 ± 0.65	0.03 ± 0.01
ImageNet-1K	66.09 ± 0.94	0.01 ± 0.01	65.42 ± 1.48	0.00 ± 0.00

C More Large Scale Experiments

C.1 Limitations of clean data fine-tuning

As noted in the main text, large-scale fine-tuning can cause models to forget broader knowledge. Table 12 shows performance on SUN397 and CIFAR100 to assess the impact of backdooring and the clean-data baselines from Table 3. Clean-data fine-tuning significantly degrades accuracy on these tasks, while `TBAR` has only a minor effect.

C.2 Impact of trigger set size

To assess the influence of the trigger set size in known-trigger scenarios (i.e., the second set of Table 3), we conduct fine-tuning experiments with varying trigger set sizes and evaluate the performance of `TBAR` vectors after one epoch. Results are shown in Figure 13, interestingly, we observe that increasing the size of the trigger set does not result in a clear performance improvement. Reinforcing the notion that the complexity of unlearning is more closely tied to the precise identification of *what* needs to be removed, rather than the scale of data.

C.3 Enhancing removal robustness with weak trigger cues

We additionally provide results on removing sinusoidal (SIG) attack (Barni et al., 2019) on ViT-B/32. In the latter case, we observed that probing the backdoored model with a reverse-engineered SIG patch consistently resulted in the label "television". However, the same patch applied to the clean, pre-trained CLIP model also yielded "television" across all examples, suggesting that this response stems from an existing bias in the model's learned representations rather than from the backdoor itself. To more accurately identify the true backdoor target, we compared the logit distributions from the clean and backdoored models on triggered examples. The class with the largest shift in density was indeed the "banana" class. This suggests that the reverse-engineered patch does not directly activate the backdoor behavior

Table 13: Results on CLIP ViT-B/32 with SIG attack, showing (CA ↑) and (ASR ↓) performance evaluated on the ImageNet-1K validation set.

	SIG	
	CA	ASR
Zero-Shot	63.34%	00.00%
Backdoored	61.36%	99.01%
Contrastive-FT	51.46%	10.26%
RoCLIP	52.61%	04.34%
CleanCLIP	51.12%	05.51%
GA	58.25%	00.10%
`TBAR`	59.02%	00.42%
GA+DECREE	56.52%	03.01%
`TBAR`+DECREE	55.41%	05.43%

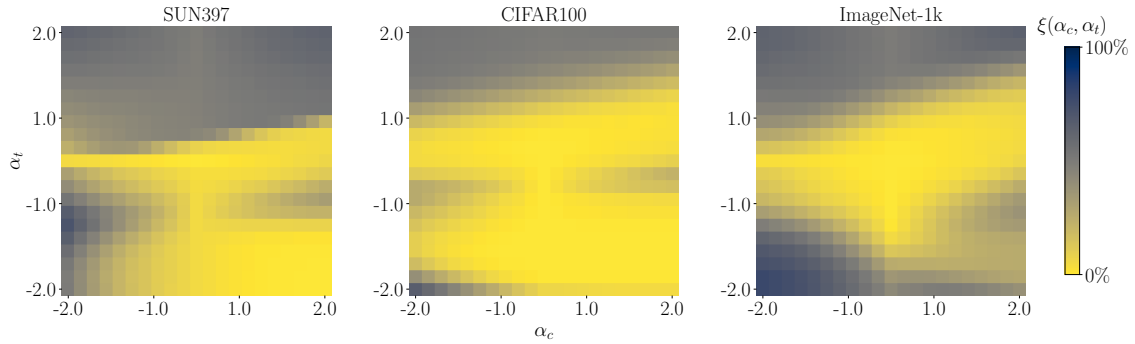

Figure 11: Weight disentanglement between clean and triggered tasks. We estimate the triggered direction $\hat{\boldsymbol{\tau}}_t$ from the backdoored model and define the clean direction $\hat{\boldsymbol{\tau}}_c$ as the residual after negation. The plots show the disentanglement error $\xi(\alpha_c, \alpha_t)$ between these task vectors, following (Ortiz-Jimenez et al., 2024). Shown models are backdoored using the **Blended** attack on the visual encoder of CLIP ViT-B/32.

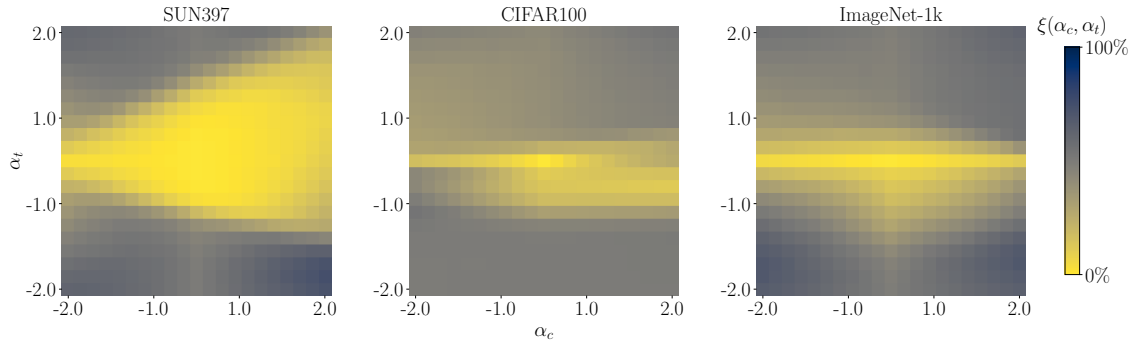

Figure 12: Weight disentanglement between clean and triggered tasks. We estimate the triggered direction $\hat{\boldsymbol{\tau}}_t$ from the backdoored model and define the clean direction $\hat{\boldsymbol{\tau}}_c$ as the residual after negation. The plots show the disentanglement error $\xi(\alpha_c, \alpha_t)$ between these task vectors, following (Ortiz-Jimenez et al., 2024). Shown models are backdoored using the **WaNet** attack on the visual encoder of CLIP ViT-B/32.

at the output level but still reveals its influence in the model's internal scoring. This observation leads to important insights. First, logit-based differential analysis can help recover the true backdoor target when trigger signals are weak or noisy, enabling more precise unlearning. Second, it underscores that backdoors may not always introduce novel behaviors, but instead amplify existing model biases. This logit difference test was additionally evaluated and confirmed for all the experiments reported in the main text.

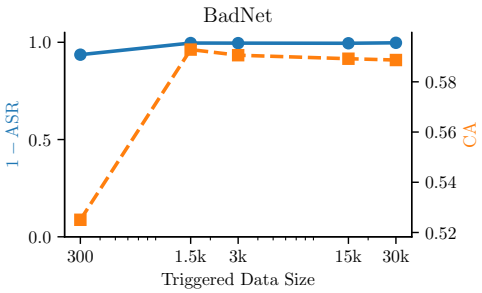

Figure 13: Impact of trigger set size on backdoor removal. Performance of `TBAR` when removing a BadNet attack from a CLIP ViT-B/32 model backdoored with image-caption data

Table 10: Controlled experiments showing the effectiveness of TBAR on single-task CLIP ConvNeXt-Base classifiers under three backdoor attacks. Clean Accuracy (CA ↑) and Attack Success Rate (ASR↓) are reported before and after sanitization.

Dataset	**CA**	**ASR**	**CA** (TBAR)	**ASR** (TBAR)
		BadNet		
CIFAR100	89.15	99.99	82.94	02.95
ImageNet	72.83	99.94	67.50	02.56
SUN397	76.99	99.99	67.48	05.11
		Blended		
CIFAR100	89.07	99.92	87.09	00.02
ImageNet	72.74	99.85	71.06	00.00
SUN397	76.89	99.93	73.21	00.00
		WaNet		
CIFAR100	89.12	99.95	86.55	00.04
ImageNet	72.78	99.99	70.67	00.01
SUN397	77.06	99.96	74.97	00.00

Table 11: Controlled experiments showing effectiveness of TBAR on transformer model (ViT-B/16) with DINO pre-training on ImageNet-1K under three backdoor attacks using CIFAR100 dataset. Clean Accuracy (CA ↑) and Attack Success Rate (ASR↓) are reported before and after sanitization.

Attack	**CA**	**ASR**	**CA** (TBAR)	**ASR** (TBAR)
BadNet	78.98	99.63	73.98	00.11
Blended	78.74	99.34	73.30	00.00
WaNet	78.38	99.08	73.43	00.04

Table 12: Out-of-distribution clean accuracy on SUN397 and CIFAR100 For CLIP ViT-B/32 model backdoored with image-caption data.

Dataset	**Pre-Trained**	**Backdoored**	**CleanCLIP**	**RoCLIP**	**Contrastive-FT**	**TBAR**
			BadNet			
SUN397	63.18%	63.23%	56.50%	58.47%	56.47%	61.47%
CIFAR100	65.58%	63.84%	48.38%	40.77%	52.39%	63.89%
			Blended			
SUN397	63.18%	63.19%	55.65%	56.43%	55.60%	62.41%
CIFAR100	65.58%	64.65%	52.31%	37.91%	52.03%	64.94%
			WaNet			
SUN397	63.18%	62.84%	56.37%	55.24%	55.66%	62.25%
CIFAR100	65.58%	62.68%	53.43%	36.32%	53.94%	61.84%